DOCUMENT RESUME

ED 262 434 CS 209 405

AUTHOR Dobrin, David N.
TITLE Some Ideas about Idea Processors.
PUB DATE May 85
NOTE 36p.; Paper presented at the Meeting of the UCLA Conference on "Computers and Writing: New Directions in Teaching and Research" (Los Angeles, CA, May 4-5, 1985).
PUB TYPE Speeches/Conference Papers (150) -- Viewpoints (120)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Cognitive Processes; *Computer Software; *Evaluation Criteria; Notetaking; *Outlining (Discourse); *Prewriting; *Word Processing; Writing (Composition); *Writing Processes
IDENTIFIERS *Idea Processors; Software Evaluation

ABSTRACT

Idea processors are computer programs that can aid the user in creating outlines by allowing the user to move, reorder, renumber, expand upon, or delete entries with a push of a button. The question is whether these programs are useful and should be offered to students. Theoretically, an idea processor prioritizes ideas by placing them in a hierarchy. Unfortunately, the idea processor does not really set priorities; it attaches labels indicating priorities that the writer has set. The confusion is between labels or symbols (which are on paper or in the computer) and ideas (which are in the head). The basic limitations of the idea processor are that (1) the meaning of an item on a list is not stable, and (2) the appropriate symbol for the meaning must change when any of the things the item depends on change. Three things need to be adjusted when a program command is used: the idea of what the item means, the idea of what other items mean, and the symbol. The question is whether having to update items is more likely to be a distraction or a way of gaining new insights. Idea processors are not likely to be useful to the ordinary user. Rather, they are likely to be useful to people who regularly plan with outlines, who tend to write things that they are sure about, and who rarely update entries in paper and pencil outlines. (HOD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

David N. Dobrin

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED262434

David N. Dobrin
Organizational Communications Consultants
1612 Cambridge St.
Cambridge MA 02138

Some Ideas About Idea Processors

"Idea processors" are computer programs which can relieve us of some of the annoyances associated with making outlines. Outlining with pencil and paper, if we want to make the branches in the tree go more than three or four levels deep, we run out of room on the page and if we wish to emend or move items, the marks that are already there get in the way. Outlining with a word processor, we can keep our copy clean, but there still isn't much room, and so we can't go very deep. Indentation on word processors, moreover, is a little tricky, and changing indentation levels or moving blocks is usually clumsy, particularly since all the labels must be updated. "Idea processors" get around all these problems. A user can move, reorder, renumber, expand upon, or delete entries with a push of a button. (A listing of the capabilities of the idea processor that I have used, called MaxThink, is given in Tables 1-0, 1-1, and 1-2.)

The existence of these programs raises the same problems for teachers of writing that the existence of word processors did a few years ago. Are the programs useful? If so, how useful? Should we offer them to students? Should their use be taught? In the rest of this article, I will try to provide some answers to these

CS209405

questions. The answers, let me say in advance, are a bit deflating. I will argue that they are not particularly useful for most people, that students don't need them and might even need to be steered away from them, and that their use certainly should not be taught.

Any casual observer might well find these conclusions counterintuitive. Planning, after all, is an important part of the writing process; outlining is frequently a useful step in planning; and therefore anything that helps people outline should be valuable. (Apparently many people find this logic so persuasive that they have invested in the programs; as I write (October, 1985) ThinkTank, the most popular of them, is seventh on the word-processing best-seller list.) Less casual observers, like the people who have developed these programs or the people who market them, might well find them ridiculous. These people are very enthusiastic about their programs; they see the programs as a major advance. Habitually, they do not characterize the programs as mere outlining aids; rather, the programs are "aids to thinking." These people give the programs names like "ThinkTank" and "MaxThink"; their generic term for the programs, "idea processors," is used without quotations marks or even any indication that quotation marks are necessary.

On the face of it, this enthusiasm is not completely

implausible. In planning a paper with an outline, you must generate ideas, decide their importance, and place them in the proper order. The programs, at the very least, help you order the ideas and designate their importance. Thus, they help you think. Anything that helps you think deserves some enthusiasm. Unfortunately, I shall argue in the next section of this article, the programs don't help you think in any interesting way. The belief that they do is based on a confusion between ideas, which are in the mind, and symbols, which are on paper. Manipulation of symbols, which is what the programs do, is not the same as manipulation of ideas; if you want to do the latter, help with the former is mostly beside the point.

The interesting question, the one that will be my primary focus in the following section is why it's beside the point. To answer that question, I will have to range over some decidedly disparate subjects, including notions of creativity and the idea that there might be something like a method for writing. Since most current theories of the writing process suggest that there is something like a method, I will not only be disagreeing with the makers of these programs but be attacking those theories. In the main, though, my approach is practical, rather than theoretical. I'll be looking carefully at the programs and trying to give you an intuitive feeling for what goes wrong when you make symbol manipulation substitute for thinking. Essentially, even though

you're saved the work of updating the symbols, you still have to update the ideas. If you don't, the fact that the updated symbols no longer correctly indicate the updated ideas makes the new outline too confusing to be useful. To use idea processors effectively, you would have to spend a lot of time learning how to cope with these confusing lists. And, except for people who are deeply committed to outlines, the time spent isn't worth while.

1. The Claims for Idea Processors

The idea processor I've used is a program called MaxThink. (A list of its capabilities is given in Tables 1-0, 1-1, and 1-2.)

This program was developed by a very energetic man named Neil Larson. Larson writes his own promotional materials and manuals, so to tell you more about his program (and, by extension, all others), I can simply quote him. According to Larson, the idea or "thought processor" is a "radical departure" from word processors or spreadsheets because it can "interact directly with higher-level thinking skills," and thus "improve the productivity of your thoughts" (P 1, 3, 1).[1] The thought-processing commands "support high-level thinking," (P 3) "expand your writing and thinking abilities," and "improve your insight, perception, imagination, and creative thinking" (P 2). The program is, in short, "mind-expanding software" (M 3).

Add Title	Begins new outline
Insert Topic	Inserts heading or subheading before or after current entry
Delete	Words or Letters
Insert Word	In current entry
Annotate	Inserts comments after entry
Jump	Moves cursor to specified path number, e.g., 1.1.3.4. This allows you to zoom in on any section of the outline
Move	Moves any item or group of items to a new area of the outline
Search and Replace	Like Word Processing

Table 1-1: MAXTHINK TEXT MANIPULATION COMMANDS

DOS File Manipulation	Renames, copies, or deletes files
Insert File	Inserts a file in an outline.
Print	Prints a file (multiple options)
View	Views a DOS file
Save	Saves your file

Table 1-2: MAXTHINK FILE MANIPULATION COMMANDS

Prioritize	Renumbers entries
Binsort	Puts entries into different "bins"
Levelize	Removes hierarchical distinctions Fence Allows you to label groups of entries
Join	Joins entries
Divide	Separates entries
Randomize	Relabels entries randomly
Sort	Sorts entries in alphabetical (or other) order
Tag	Provides a space for tags, which can then be used in sorting

MaxThink also has a LISP-like "Thought Processing Language," which allows you to program the use of these commands.

Table 1-3: SPECIAL MAXTHINK COMMANDS

In some trivial sense, these claims are true of virtually any tool; I would probably grant a carpenter's claim that a T-square is in some sense "mind-expanding" and "thought-supporting." But Larson means something stronger than this. For him, the program literally takes over some of the thinking jobs for you. "MaxThink provides the system, structures, and commands for pondering, discovering, expanding, and integrating information" (P 3).

One might, of course, make this stronger claim for any tool that performs algorithmic processes. A calculator, for instance, expands your adding capabilities and supports additive thinking by taking over some of the adding for you. Without a calculator, you must enter numbers (on a sheet of paper), add them, and enter the result. With a calculator, all you do is enter the numbers. The calculator adds and enters the result. By analogy, an idea processor like MaxThink might help you prioritize by taking your entries and placing them in a hierarchy. Unfortunately, though, the analogy is confused. The confusion is between labels or symbols (which are on paper or in the computer) and ideas (which are in the head). A priority is an idea; a statement of a task with a number 1 attached is a symbol which stands for an idea. In "prioritizing," an idea processor does not set priorities; it attaches labels which indicate priorities that the writer has set. This confusion underlies all the false claims about idea processors.

The confusion, by the way, is the same in my earlier description of how calculators work. Calculators do not manipulate numbers, which are meaningful; they manipulate symbols for numbers. Strictly speaking, they don't add; they simulate adding. Calculators simulate successfully because (for numerical calculations) we have worked out mechanical processes which manipulate symbols in a way that exactly matches the way we manipulate numbers. So the confusion doesn't matter; it makes sense to speak of the calculator as adding, because the calculator performs a process which is formally similar to addition.

If we could work out a similar mechanical processes for prioritization, then again, the confusion would not matter; we could sensibly speak of a prioritizer as takes over some of our prioritizing processes for us. But no such mechanical prioritization processes have been worked out, nor are any likely to be.[2] And in any case an idea processor is not an implementation of such a process.

That idea processors do not process ideas, but symbols and that they therefore don't support thinking processes any more than word processors or T-squares do should be completely obvious. Why then do people like Larson make these claims or (worse yet) believe them? I think they allow themselves to be confused by the difference between ideas and symbols because they make three, quite

different assumptions.

1. Thinking is fundamentally algorithmic (and thus, according to Church's and Turing's thesis, it can be simulated by a mechanical process). This assumption has been accepted as obviously true in our culture ever since Plato.[3] This assumption does not, as I have pointed out, mean anything when it comes to idea processors because they don't incorporate a mechanical process that does simulate thinking. But it's easy with this assumption to make the mistake of believing that any algorithmic process that facilitates thinking is a special kind of tool for thinking, because it works in the same way that thinking does. That's why it's easier to think of a calculator as a thinking tool than it is to think of a T-square as a thinking tool. The mistake is commonly made. The programming ideas used in MaxThink, for example, are based on programming ideas incorporated in LISP, a language developed for artificial intelligence applications. In the artificial intelligence community, LISP implementations are commonly spoken of as just such special aids to thinking. Indeed, according to one LISP instructor at M.I.T., LISP is the language that "lets a computer think."

2. Meanings are on the page, so manipulating the symbols is manipulating the meanings. Again, there's powerful support for this idea. Philosophers routinely argue that meanings are not in the head. [7] Most brands of linguistics assume (1) that there is a distinction between the meaning of a word and its use and (2) that the literal meaning is independent of the use.[4] This assumption does not, of course, mean that any manipulation of symbols is a manipulation of their meaning, because many manipulations of symbols (e.g., rearranging the letters) don't preserve the meaning through the manipulations. On this assumption, a symbol manipulator can be a meaning manipulator only if it is what John Haugeland calls a "semantic engine," [5] that is, if it preserves the meaning during each operation. There is no reason to believe, again, that idea processors are "semantic engines." But many people believe that it is possible to build a machine which manipulates symbols according to more powerful semantic and syntactic rules and which would then literally help us write. And idea processors might well be the first step on the way.

3. The best method of writing is to make an outline of the paper to be written and then expand the entries

> indefinitely. If you believe this, whether or not you believe the other assumptions, you should use idea processors, because they make this kind of writing easier (apparently). People who accept this assumption, surely, are the people making ThinkTank into a best-seller.

Combine one, two, and three, add a pinch of confusion and a tablespoon of belief that your livelihood is on the line, and you can see why people believe that idea processors are accurately named. Add for dessert the belief that "most mental processes simply amount to picking the best option from a group of many choices" (P 3), and you can get up from the table thinking that idea processors ought to be on everyone's desk.

If you don't believe these assumptions, then suddenly the programs don't look so appetizing. The question then becomes "How good are they really?" In the next section, I want to address that question.

2. How Useful Are Idea Processors

If idea processors only manipulate symbols of ideas, allowing one to group them, expand upon them, reorder them, put them in hierarchies, and (alas) erase them, then are they useful in planning, noodling, brainstorming, or "shifting perceptions," as

the makers claim, and are they as useful in writing as those who believe the last assumption apparently believe? And if so, how?

The answer is complicated, because any tool can be useful, if you are committed to using it for some purpose and you want to take the time to learn how. You can probably, eventually, cut diamonds with a backhoe. So I'm not going to say that idea processors are not useful, only that in normal situations, they're not as useful as they might appear. In making this evaluation, I can talk only about normal, reasonable use, and even then I will probably get the association of backhoe owners mad at me.

Hence, to show you how and why these programs fail to be useful, I am simply going to describe their basic limitations and then ask you to get an intuitive grasp of how those limitations work by following some simple examples. The basic limitations are these:

1. The meaning of an item in a list is not stable. It depends on the other items in the list, the reason the item appeared, facts about the item that are known to be relevant, the purposes of the author, etc., etc.

2. The appropriate symbol for the meaning must change when any of the things the item depends on change.

These limitations have a very simple consequence: to use the programs effectively, you have to be constantly updating the lists. Since the meaning of all the items changes whenever you move them around, change your purposes, etc., you can't change the lists with any precision by using the program's commands alone. Whenever you use a program command, you also have to adjust three things: 1) your idea of what the item means, 2) your idea of what other items mean, and 3) the symbol.

An example will help. I am in the habit of making lists of things to do. When I first got MaxThink, I thought I could make my life much simpler by putting all these lists together. Say I had a list of things to do this week, like that shown in Table 2-0. I now want to compile a new list, things to do today. If I use an idea processor, the best I can get easily is something like the list shown in Table 2-1. What I want, of course, is something like Table 2-2.

This is, I admit, a simple-minded point, and in this form, it doesn't seem to be too serious an objection. I can, after all, change the entries as I move them, or if I am too lazy to do that, I can remember that, in the new list, the meaning of (e.g.) buy groceries is now different.

When list processing gets more complicated, however, it's not

Things To Do This Week

A. Work on Idea Processing Paper

B. Buy groceries

C. Do laundry

D. Write letter to AE

E. Grade papers for Technical Writing Class

Table 2-1: A LIST ENTERED ON MAXTHINK

Things To Do Today

A. Work on Idea Processing Paper

B. Buy groceries

C. Grade papers for Technical Writing Class

Table 2-2: A LIST CONSTRUCTED BY MOVING ENTRIES FROM THE OLD LIST

Things To Do Today

Develop Second Section more fully

Buy cucumbers

Grade 3 (?) papers

Table 2-3: THE NEW LIST WITH THE ENTRIES MODIFIED APPROPRIATELY

easy to do either of these things. For a big list, updating is a tremendous chore. Yet failing to update requires that I remember the new (or old, depending) meaning, and I, at least, have a very hard time doing that. To show you how really difficult this problem is, let me show you another example, an entirely immediate one. Let us look at an outline of the overall structure of this paper and compare it to an outline of this section of the paper. I gave you an outline of the overall structure at the end of the introductory section. In Table 2-4, I reproduce it. In Table 2-4 I give a list of things I'm talking about right now, essentially my outline for this page and the pages surrounding it. Now, try to fit the second list into the first list. It's not, as you can see, at all easy. Does this mean that I'm writing badly, that I'm straying from my outline? You can determine the answer by running the following little test. Try to remember whether you thought I was straying before you happened on this page. If you did, perhaps I was. If you didn't, then you are grasping the structure of this, even though it's difficult for you to explain it. Most readers of the drafts had no problems.

There is an important point about the writing and reading processes buried in this simple example. Most modern theorists of language comprehension believe that in any comprehensible piece of prose, you are setting up a master outline of the piece, and hanging each idea somewhere in that outline as you read.[5] Where

I. Look at the claims.

II. Look at the programs

A. Practical evaluation.

B. Not too useful.

1. Unless you want to put a lot of time in them.

Table 2-4: LIST OF THINGS I SAID I WOULD DISCUSS

1. Anything can be useful.
2. Look at limitations.
    2.1. Appropriate meaning is contextually determined.
    2.2 Appropriate symbol is contextually determined.
3. Meanings are unstable.
4. Examples.
    4.1 Grocery example.
    4.2 Paper example.
    4.3 Lions-tigers list example.

Table 2-5: LIST OF THINGS I'M NOW TALKING ABOUT

you don't understand, it's because the master outline isn't set up right. So if the theory is right and the piece of prose is good, you shouldn't have any problem putting the outlines together, because that's what you do anyway, as you read. My little experiment, then, suggests that this theory is wrong. If most people have no trouble reading this kind of writing but do have trouble constructing outlines of it, that is a counterexample to the cognitive thesis, one that must be explained.

I have a non-cognitive explanation of why it is so difficult to meld the two lists. I think that in understanding each list, we do more than work with the symbols on it. We supply and we work with the relationships among the items, the purposes in making the list, the ideas not mentioned but relevant, the habits of evaluating relevance, and, of course, the meanings of the items on the list. All this material we supply makes up a kind of mental glue, which holds together the items on the lists and makes them sensible. When we try to conflate the two lists, we have to break the glue that held together the items on each and then try to find new glue.[6] It is that glue plus our intentions which makes up the meaning we give each item on the list; it is that glue which makes meaning context-sensitive.[7] Manipulations of the symbols become manipulations of meaning only when the glue is shifted, too. The computer program, however, only conflates the symbols.

Again, it's easier to see if you have an example. My argument is that rearranging the items on a list can change the meaning of the items on the list. The reason is that a huge array of ideas, memories, purposes, etc., are helping to determine that precisely this meaning is meant in this list, and that the precise operation of this array is incredibly sensitive to context. I can show you that this is true by showing you a list of two items, changing the context slightly, and showing you that the meaning of the items on the lists changes. Consider the following pair of items.

Lions

Tigers

Consider what happens when an item is added.

Lions

Tigers

Wolves

This is, let us say, a list of Asian carnivores. Now subtract the last item and add a different one.

Lions

Tigers

Pistons

This is a list of Detroit's professional sports teams. Again, change the last item.

Lions

Tigers

Bears

Oh, my.

Even changing the title can make a huge difference. If I were to take the first list and call it "Large North American Carnivores," both the meaning of each item and the glue would shift once more. There would be the suggestion now that all the animals are in the same ecosystem.

You can see now why people might have a hard time if they wanted to write a paper or report by expanding an outline. As they expand the outline, the meaning of the entries, the relationships between the entries, and the appropriate symbol for an entry all shift. We know that intuitively when we try to write from an

outline, because we find that as we write, headings become subheadings, topics get split up or eliminated, ideas or data that we thought were telling become defanged. The reason, of course, is that the context, our purposes, the world of relevant knowledge, etc. have all changed, and so the meaning of the entries changes.

The problem, then, with idea processors is that they are insensitive to changes in meaning. How serious a problem is that? Consider the list of MaxThink commands in Table 1-1. Ask yourself how many of these commands would be useful if you were trying to work with large blocks of text or many-itemed lists. Surely not any of the advanced commands. Even the simple commands, the ones that made it easy to make an outline, will interfere when you are trying to turn the outline into a text or when you're trying to update many entries.

3. It's Not a Bug, but a Feature

There is a simple reply to these objections. If in fact manipulating the outline requires that we update it, and if in fact updating it requires that we rethink what we're doing (which groceries?, Aha, cucumbers), then using an idea processor forces us to rethink. Among enthusiasts, this is not a bug, but a feature, a terrific feature. Again, let me quote Larson. Using MaxThink, the promotional material says, allows you to perform "careful mental exploration of the boundaries of your . . . information" (P 3). It

"lets you purposely shift your perspectives to gain as much information as possible." Using it "enables you to shift your viewpoints to bypass your current perceptions and attain additional insights."

Larson is clearly right, in a sense. Any time we are rethinking our meaning, new ideas may occur to us, gaps in our reasoning may appear, and so on and so forth. This is what people like Ann Berthoff are trying to get at when they claim that writing is thinking.[8] When you write, by putting things down in words and so objectifying what you have to say, you discover new relationships among the ideas. So, if you believe people like Berthoff, you have to believe people like Larson, at least a little bit, since updating outlines is vaguely similar to (re)writing.

This account, unfortunately, is just too vague about what thinking is. True, we get new ideas by writing or by updating outline entries, but we would also get new ideas by contemplating our navel or by getting in our car and driving around. The question in evaluating this or any other claim about things that help us think is whether they usually help us direct our thoughts in productive ways. Thus writing, we are inclined to say, is a productive way of improving our thinking when we're trying to make an argument or get clear about an abstract argument because for some odd reason, when we write out our ideas, germane objections,

relevant side issues, and felicitous ways of expressing ourselves occur to us. Contemplating our navel(s?), on the other hand, is simply less productive in the same situation (again, for most of us), and driving is (often) counterproductive. In the case of writing, we can even intuitively see how our thinking might be productive, and in the case of contemplating our navel, we can see how it might not be. So the question to ask about idea processors is not whether they can help us think, but whether having to update items is more likely to be a distraction (forcing unproductive, useless patterns of thought on us) or a way of gaining new insights.

Well we've got the right question now, and I think if you're with me at all, you know what I think the answer is. But I can't prove it. The trouble is twofold. First, in any creative endeavor, you have to keep a balance between working in a directed way toward the goal and relaxing a bit, trying to see things in a new light or getting a new idea from left field. Sometimes it's better to go for a drive. Second, in any creative endeavor, people can learn to use almost any method in ways that make them productive. Some people can only get ideas when they go for a drive. So idea processors could be used as a way to relax, or they could be a way of being directed, an inadequate way generally, but a way which practice and adaptation make adequate for some people.

I can't recommend that you learn how to make a idea processor be a tool for directed thinking, because it's obviously the wrong sort of tool. But perhaps it could be a handy tool for non-directed thinking. Perhaps a convenient outline-maker allows us some free play of ideas which we can usually find useful. Perhaps, indeed. But it does seem to me that there are several reasons to believe that this doesn't happen. The most compelling to me is that it doesn't happen when I use the program. More compelling to you might be the reason why.

When I do try to look at what I've done, searching for new ideas, holes, a different viewpoint, etc., I find that I have to range widely over all of it. I find, for instance, that when I review notes, it's helpful to have all the notes all over the table, so that I can move quickly from one to the other. If we take the notes as a metaphor for the mind, I find that the items on the fringe of my (visual, mental) horizon change the shape, aspect, viewpoint, etc., of the rest of what I see. One little idea, something I'm not even paying attention to, out there on the fringes of my note cards, may cause me to see what I'm looking at in a new light. This is, by the way, a normal experience in other contexts; it's well-known that in ambiguous pictures (like Necker cubes), changing the background even slightly changes the way we see the picture. I'm arguing that it's also normal here.

If that is the case, then a feature of a thought-supporting tool (putting note cards all over the place or idea processors) that impedes our ranging freely is a bug, not a feature. In pencil and paper outlines the fact that you have to mess up the page in order to reorder items is a bug, and the fact that you can't go several levels is another bug. In idea processors, there's an equivalent bug. In idea processors, you can only see two levels, a heading and subheadings under that heading, at any one time. You can't, in other words, ever range freely over the headings on the computer screen. If you're working with a four-deep tree, you may have quite a time even finding another entry which might be relevant.

Unfortunately, this is a very serious bug. If the meaning of any entry is incredibly sensitive to context, and much of the context that can be presumed to be relevant is simply invisible, then any work you do do with the entries in front of you can be made nugatory as soon as almost any other entry hoves into view. Paper and pencil outlines may be clumsy because they're hard to update, but at least while you're working with them, you can update in a relatively purposive way, because all the relevant information is visible. With paper and pencil outlines, moreover, you have more ways of representing relevant relationships. You can underline, draw arrows, circle, shade, highlight, cross-out, etc., etc. With such tools, you can make very delicate adjustments in

the context, and thus range over the ideas with rather fine purposes. "Aha, this goes here, and oh, I might bring in a shaded reference to that over in this spot" can all be represented with an arrow, but it can't even be thought with an idea processor.

An idea processor, in other words, is likely to make both purposive exploration or of ideas or free play with ideas more difficult rather than less. True, it makes manipulation of entries easier, but it does this by hiding the context, and it is the context, the fringes of consciousness, which must be made available if free-ranging is going to be productive. One can, as I said, learn how to use idea processors for these purposes. But there is no a priori reason to think that they are designed for or suited to this kind of exploration, no reason to think that they are precision cutting tools rather than backhoes.

4. Some Morals of the Story

Can idea processors be useful? Yes, no question about it. Are they likely to be useful to the ordinary user? No. Are they likely to be useful to people who regularly plan with outlines, who tend to write things that they're sure about, and who rarely update entries in paper and pencil outlines? Yes. Will idea processors make writing easier by making writing from outlines easier? No, since the usefulness of writing from outlines is crucially dependent on having the whole outline in front of you, so that you

can understand each item in the relevant context. Will idea processors facilitate a writing process in which people write by expanding each outlined item easier? Yes, and unless they are very skilled in using what I consider to be an ineffective way of planning and writing, it will, in effect, confirm them in their bad habits. Should you use idea processors? Who knows. If you think you might, try investing $65.00 in MaxThink.

So much is clear, and, I hope, unexceptionable. I could stop here. Before I quit, however, I want to bring out some other consequences of this discussion. If my observations about the weaknesses of the idea processor have any validity at all, then they pose two, previously unnoticed problems for modern accounts of the writing process. The first problem is that the writing process on my account has at least two, distinct steps. In the first, we might put down symbols for ideas and give them some kind of provisional relation; we are gathering what we have to say and allocating a position in the hierarchy to each item. That stage, in classic accounts, is the outlining stage. In the second, we expand upon those ideas, make them more precise and detailed, and put them into more precise relation. This relation is not a more precise definition of order and importance (which is what the outline gave us); the second step does not hang each idea on a more precisely-defined branch. Rather, in the second step we give the ideas different kinds of relations, those offered us by the endless

resources of English prose. Relationships of order and importance fall by the wayside. That stage, in the classic accounts, is the writing stage.

I am not suggesting that we return to the classic accounts. But I would like to point out that such authoritative accounts as those given by Linda Flower and John R. Hayes do not make any distinction between the two steps. [4, 3] Flower and Hayes describe formal or procedural processes for solving writing problems which would apply equally well whether we are outlining or writing. Yet the simple fact that idea processors collapse for any normal user (namely me) who wishes to write by expanding the entries suggests that the procedures need to be different. Perhaps, as my discussion suggests, the more finely-grained the context you know to be relevant (what you've written before), the thicker the glue, and the less useful it is to do formal manipulations.

The second problem is that the first step in the writing process seems to be more free-ranging than her account suggests. Flower (and all other cognitivists) describe writing as a problem-solving process. [4] This suggests that once a problem is solved, it's solved, so that one can (usually) work on things piecemeal. But if the failure of idea processors is that they don't allow you to range freely in the early going, that all

relationships among ideas are flexible then and open to question, that almost any information might be relevant to any writing decision, and that many different activities, including contemplating one's navel, might be helpful, then the writing process in the early going stops looking like a problem-solving process, because it isn't closed, and it doesn't move steadily toward a solution.

If I am right, then idea processors do have one use: to study the writing process. If I am right, the way they interfere with people's purposes during the writing process would tell us something interesting about what the people are trying to do. Such a study would have to recognize the vast differences in individual styles, and it would have to recognize, too, how determined by the actual meaning of what is written any writing process is. If these were accurately taken into account (no mean task), then this somewhat impressionistic, personal, and intuitive criticism of these programs could either be confirmed or confounded. More important, the results could teach us something new about a completely mysterious, endlessly fascinating subject, the writing process.

REFERENCES

1. Berthoff, Ann. Forming/Thinking/Writing: The Composing Imagination. Boynton/Cook Publishers, Upper Montclair, N.J., 1978.

2. Dreyfus, H.. What Computers Can't Do. Harper and Row, New York, 1979. rev. ed.

3. Flower, Linda and Hayes, John R. Identifying the Organization of Writing Processes. In Cognitive Processes in Writing, Erlbaum, Hillsdale, N. J., 1980.

4. Flower, Linda and Hayes, John R. "A Cognitive Process Theory of Writing". College Composition and Communication 32 (Dec. 1981), 359-382.

5. Haugeland, John. Semantic Engines. In Mind Design, Haugeland, John, Ed., MIT Press, Cambridge, Ma., 1981.

6. Kintsch, Walter. The Representation of Meaning in Memory. Erlbaum, Hillsdale, N. J., 1974.

7. Putnam, Hilary. The Meaning of Meaning. In Collected Papers, Harvard University Press, Cambridge, Mass., 1984.

8. Searle, J. R. Literal Meaning. In Expression and Meaning, Cambridge University Press, Cambridge, 1979.

9. Searle, John R.. Intentionality. Cambridge University Press, Cambridge, 1983.

Notes

[1] I am quoting the promotional material for MaxThink (P), from the MaxThink Newsletter (N), and from the MaxThink manual (M). All are published by the MaxThink Corporation in Piedmont, California; all are circa 1984. References for the quotations will give the letter and page number, where pages are numbered.

[2] See my "Limitations on the Use of Computers in Writing and

the Teaching of Writing" in The Future of Computers in Composition, forthcoming. For a full discussion of the limitations of computers, see Hubert L. Dreyfus, What Computers Can't Do.

[3] For more on this, see Dreyfus, op. cit. and also Putting Computers in their Proper Place, forthcoming.

[4] I follow John Searle in rejecting these ideas about meaning. See "Literal Meaning," Expression and Meaning (Cambridge: Cambridge University Press, 1979) and "Are Meanings in the Head?" Intentionality (Cambridge: Cambridge University Press, 1983).

[5] This is one way of putting the underlying thesis of cognitive psychology. The essential idea is that people process and store information by using list structures. An outline is a list structure. For a typical example, see Kintsch [6]

[6] There is no standard set of technical terms with which I can describe this phenomenon. In phenomenology, the items supplied which allow us to understand whatever it is we're actively aware of are in what is called a "horizon of consciousness"--the analogy is to the horizon of our vision--but there is no adequate account of how we have to rearrange that horizon in order to make two quite disparate things make sense in terms of each other.

[7] In this last sentence, the glue is what John Searle, Hubert L. Dreyfus, and Ludwig Wittgenstein call "the Background." See John R. Searle, Intentionality (Cambridge: Cambridge University Press, 1983).

[8] See Forming/Thinking/Writing: The Composing Imagination. Not that Berthoff would approve of idea processors.